07.03.00

A

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b)

Attorney Docket No. 042390.P5563

First Named Inventor or Application Identifier Padwekar

Express Mail Label No. EL591668250US

ADDRESS TO:	**Assistant Commissioner for Patents** **Box Patent Application** **Washington, D. C. 20231**

APPLICATION ELEMENTS
See MPEP chapter 600 concerning utility patent application contents.

1. ____ Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)

2. _X_ Specification (Total Pages 23)
(preferred arrangement set forth below)
- Descriptive Title of the Invention
- Cross References to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to Microfiche Appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings (if filed)
- Detailed Description
- Claims
- Abstract of the Disclosure

3. _X_ Drawings(s) (35 USC 113) (Total Sheets 7)

4. ____ Oath or Declaration/Power of Attorney (Total Pages ____)
   a. ___ Newly Executed (Original or Copy)
   b. ___ Copy from a Prior Application (37 CFR 1.63(d))
   (for Continuation/Divisional with Box 17 completed) **(Note Box 5 below)**
   i. ___ DELETIONS OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).

5. ____ Incorporation By Reference (useable if Box 4b is checked)
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

6. ____ Microfiche Computer Program (Appendix)

7. ____ Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)
   a. ___ Computer Readable Copy
   b. ___ Paper Copy (identical to computer copy)
   c. ___ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. _____ Assignment Papers (cover sheet & documents(s))

9. _____ 37 CFR 3.73(b) Statement (where there is an assignee)

10. _____ English Translation Document (if applicable)

11. _____ a. Information Disclosure Statement (IDS)/PTO-1449

_____ b. Copies of IDS Citations

12. _____ Preliminary Amendment

13. _____ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)

14. _____ a. Small Entity Statement(s)

_____ b. Statement filed in prior application, Status still proper and desired

15. _____ Certified Copy of Priority Document(s) (if foreign priority is claimed)

16. X Other: Unsigned Declaration and Power of Attorney (5 pages)

17. **If a CONTINUING APPLICATION,** check appropriate box and supply the requisite information:

___ Continuation ____ Divisional ____ Continuation-in-part (CIP)

of prior application No: _____

18. **Correspondence Address**

_____ Customer Number or Bar Code Label _____ (Insert Customer No. or Attach Bar Code Label here)

or

X Correspondence Address Below

NAME Michael A. DeSanctis Reg. No. 39,957 6/30/2000

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

12400 Wilshire Boulevard 7th Floor, Los Angeles, California 90025

(303) 740-1980 Telephone (303) 740-6962 Facsimile

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number: EL591668250US

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, DC 20231.

June 30, 2000
Date of Deposit

Name of Person Mailing Correspondence

Heather South 6/30/00
Signature Date

Serial/Patent No.: — Filing/Issue Date: —
Client: Intel Corporation
Title: A Branch Target Buffer (BTB) Including a Speculative BTB (SBTB) and an Architectural BTB (A-BTB)
BSTZ File No.: 042390.P5563 Atty/Secty Initials: MAD/hss
Date Mailed: June 30, 2000 Docket Due Date: —

The following has been received in the U.S. Patent & Trademark Office on the date stamped hereon:

- ☐ Amendment/Response (____ pgs.)
- ☐ Appeal Brief (____ pgs.) (in triplicate)
- ■ Application - Utility (23 pgs., with cover and abstract)
- ☐ Application - Rule 1.53(b) Continuation (____ pgs.)
- ☐ Application - Rule 1.53(b) Divisional (____ pgs.)
- ☐ Application - Rule 1.53(b) CIP (____ pgs.)
- ☐ Application - Rule 1.53(d) CPA Transmittal (____ pgs.)
- ☐ Application - Design (____ pgs)
- ☐ Application - PCT (____ pgs.)
- ☐ Application - Provisional (____ pgs.)
- ☐ Assignment and Cover Sheet
- ■ Certificate of Mailing
- ■ Declaration & POA (5 pgs.) (unsigned)
- ☐ Disclosure Docs & Orig & Copy of Inventor's Signed Letter(____pgs)
- ■ Drawings: 7 # of sheets includes 7 figures
- ■ Express Mail No.: EL591668250US ☐ Check No.____
- ☐ ____ Month(s) Extension of Time Amt: ____
- ☐ Information Disclosure Statement & PTO-1449 (__ pgs.) ☐ Check No.____
- ☐ Issue Fee Transmittal Amt: ____
- ☐ Notice of Appeal
- ☐ Petition for Extension of Time
- ☐ Petition for ____
- ■ Postcard
- ☐ Power of Attorney (____ pgs.)
- ☐ Preliminary Amendment (____ pgs.)
- ☐ Reply Brief (____ pgs.)
- ☐ Response to Notice of Missing Parts
- ☐ Small Entity Declaration for Indep. Inventor/Small Business
- ■ Transmittal Letter, in duplicate
- ☐ Fee Transmittal, in duplicate

☐ Other: ____

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number: EL591668250US

I hereby certify that I am causing the above-referenced correspondence to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated below and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

June 30, 2000
Date of Deposit

Heather S. South
Name of Person Mailing Correspondence

Signature 6/30/00
Date

Attorney Docket No.: 042390.P5563 Patent Application

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

TITLE OF THE INVENTION

A BRANCH TARGET BUFFER (BTB) INCLUDING A SPECULATIVE BTB (SBTB) AND AN ARCHITECTURAL BTB (ABTB)

INVENTOR

KIRAN A. PADWEKAR
1520 VISTA CLUB CIRCLE #203
SANTA CLARA, CA 95054

Prepared by

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(303) 740-1980

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EL591668250US
Date of Deposit: June 30, 2000
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner of Patents and Trademarks, Washington, D. C. 20231

Heather S. South
(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

6/30/00
(Date signed)

A BRANCH TARGET BUFFER (BTB) INCLUDING A SPECULATIVE BTB (SBTB) AND AN ARCHITECTURAL BTB (ABTB)

COPYRIGHT NOTICE

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates generally to the field of branch prediction. More specifically, the invention relates to the use of a Speculative Branch Target Buffer (SBTB) to maintain speculative branch data for in-flight branches.

Description of the Related Art

Early microprocessors generally processed instructions one at a time. Each instruction was processed using separate sequential stages (e.g., instruction fetch, instruction decode, execute, and result writeback). Within such microprocessors different dedicated logic blocks performed each different processing stage. Each logic block waited until all the previous logic blocks completed operations before beginning its operation

To improve efficiency, microprocessor designers overlapped the operations of the logic blocks for the instruction processing stages such that the microprocessor operated

on several instructions simultaneously. In operation, the logic blocks and hence the corresponding instruction processing stages concurrently process different instructions. At each clock tick, the result of each processing stage is passed to the subsequent processing stage. Microprocessors that use the technique of overlapping instruction processing stages are known as "pipelined" microprocessors. Some microprocessors further divide each processing stage into substages for additional performance improvement. Such processors are referred to as "deeply pipelined" microprocessors.

An example of a simplified instruction pipeline 100 is shown in Figure 1. According to this simplified example, the instruction pipeline 100 comprises five major stages 105-130. The five major stages are the fetch stage 105, the decode stage 110, the dispatch stage 115, the execute stage 120, and the writeback stage (also referred to as the retirement stage) 125. Briefly, during the first stage, the fetch stage 105, one or more instructions are retrieved from memory, and subsequently decoded into micro-ops during the decode stage 110. Then, the micro-ops are dispatched to the appropriate execution unit for execution during the dispatch stage 115 and execution takes place during the execute stage 120. Finally, as the micro-ops complete execution, they are marked as being ready for retirement and are subsequently retired (e.g., their results are committed to the architectural registers) during the retirement stage 125. Consequently, the fetch unit (not shown) at the head of the pipeline provides the pipeline with a continuous flow of instructions, hence keeping the microprocessor busy. The fetch unit keeps the constant flow of instructions so the microprocessor does not have to stop its execution to fetch an instruction from memory. Such fetching guarantees continuous execution, as long as the instructions are stored in order of execution. However, due to branch instructions, such as conditional branch instructions included in software loops or conditional jumps,

instructions encountered by the fetch unit are not always presented in a sequence corresponding to the order of execution. Thus, branch instructions can cause pipelined microprocessors to speculatively execute down the wrong path such that the microprocessor must later flush the speculatively executed instructions and restart at a corrected address.

As a result, many pipelined microprocessors employ branch prediction techniques to predict the outcome of branch instructions (e.g., determine which instruction to fetch next). Generally speaking, branch prediction seeks to guess whether or not a branch encountered in the instruction stream will be taken or not; and to fetch executable code from the appropriate location in the instruction stream. When a branch instruction is executed, it and the branch target address (i.e., the address of the of the instruction to be executed if the branch is taken) are stored in a branch target buffer (BTB). This and other information is subsequently used to predict which way the instruction will branch the next time it is executed. Mispredicted branches still cause the instruction pipeline to stall while the incorrect sequence of instructions that have been fetched and have begun execution are flushed from the instruction pipeline. However, when the branch prediction is correct (as it is over 90 percent of the time), executing a branch does not cause a pipeline stall as the processor may fetch and begin executing the proper sequence of instructions in advance.

An earlier branch target buffer cache implementation is illustrated in Figures 2 and 3. The branch target buffer (BTB) 200 depicted in Figure 2 is a set-associative cache that stores information about branch instructions in 128 individual "lines" of branch information. Each line of branch information in the BTB 200 contains four branch entries that each contains information about a single branch instruction that the

microprocessor has previously executed (if the valid bit is set in the entry). Each line also includes a branch pattern table 221 and least recently replaced (LRR) bits 220. The branch pattern table 221 is used for predicting the outcome of conditional branch instructions in the line of branch entries. The LLR bits 220 are used by the branch prediction circuit to select a branch entry in the line when information about a new branch will be written into the line of branch entries.

Figure 3 illustrates the branch information stored within each branch entry of the BTB 200. As illustrated in Figure 3, each branch entry contains a tag field 310, a block offset field 320, a branch type field 330, a true history field 340, a speculative history field 350, a history selection bit 370, a valid bit 380, and a branch target address field 390. The tag address 310 and the block offset 320 are used to identify a memory address of the branch instruction associated with the branch entry. The branch type field 330 specifies what type of branch instruction the branch entry identifies (e.g., conditional branch, return from subroutine, call subroutine, unconditional branch). The true history field 340 maintains the actual (fully-resolved) taken or not-taken history of the branch instruction for a predetermined number of prior executions. The speculative history field 350 maintains the "speculative" taken or not-taken history of the branch instruction for the predetermined number of prior executions. The history selection bit 370 indicates which of the true history field 340 or the speculative history field will be used to index into a pattern state table when calculating a branch prediction. The valid bit 380 indicates whether or not the branch entry contains valid branch information. The valid bit 380 is typically set during the execute or retirement stage when the branch prediction circuit allocates and fills the corresponding branch entry. The valid bit 380 is cleared when the branch entry is subsequently deallocated by the branch prediction circuit.

Because many of the fields (e.g., tag 310, valid 380, block offset 320, LRR 220, pattern table 221, true history 340, and speculative history 350) of the BTB 200 must be accessed by various pipeline stages the BTB 200 must include multiple ports for reading/writing the appropriate fields at prediction time and reading/writing the appropriate fields during allocation, update, and deallocation of branch entries.

In such a prior BTB 200, branch entries are typically allocated at execute or retire time to avoid allocating entries along a mispredicted path. This, however, results in mispredicting tight loops until they are allocated. For deallocation, two consecutive lines of instruction are deallocated when a bogus branch is encountered, resulting in deallocation of good branches. Finally, branches are typically updated at execute time instead of retirement to improve prediction. This, however, often results in corruption since not all executed branches retire.

BRIEF DESCRIPTION OF THE DRAWINGS

The appended claims set forth the features of the invention with particularity. The invention, together with its advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

Figure 1 illustrates a simplified instruction pipeline.

Figure 2 illustrates a prior art branch target buffer (BTB) implementation.

Figure 3 illustrates branch information stored within each branch entry of the BTB of Figure 2.

Figure 4A is a block diagram of a computer system in which one embodiment of the present invention may be implemented.

Figure 4B is a simplified block diagram of various microprocessor units that may interact with the branch prediction circuit of the present invention.

Figure 5 is a simplified block diagram of a branch prediction circuit according to one embodiment of the present invention.

Figure 6 is a flow diagram illustrating branch entry processing according to one embodiment of the present invention.

DETAILED DESCRIPTION

A method and apparatus are described for improving the performance of branch prediction using a combination of speculative branch target buffer and architectural branch target buffer. According to one embodiment, a branch target buffer includes both a speculative branch target buffer (SBTB) and an architectural branch target buffer (ABTB). The SBTB may be implemented as a relatively small structure that supports the ABTB, and that can be used to maintain speculative branch data for in-flight branches (i.e., those that have been fetched but not yet retired). Thus, the ABTB need only store the architectural or actual branch data. The combination of ABTB and SBTB described herein seek to improve the cost and performance of branch prediction, which essentially lowers cost and improves performance of a microprocessor.

According to one embodiment, the SBTB allows the speculative history and the selection bit to be eliminated from the ABTB, and allows the ABTB to be single-ported, saving area that can be traded for performance. As will be described further below, branches can be allocated speculatively in SBTB at the time of decode, helping avoid misprediction in tight loop branches. Bogus branches are also deallocated at decode time. They are deallocated in the line containing the branch, and the next line only if it is a consecutive line thereby eliminating unnecessary deallocation.

The branch entry is updated speculatively at prediction time, and corrected at execution time in the SBTB, thereby reducing the number of ABTB accesses. Further, the branches may be updated in the ABTB only after the last of the branches in the line retire to reduce update traffic to the ABTB. Both of these make a single-ported ABTB possible. Finally, there is no corruption of branch data as a result of mispredicted branches because the update is at retire time.

According to one embodiment of the present invention, the method and apparatus consist of a SBTB having all entries searched in parallel to determine whether the set matches against a fetch instruction pointer (IP). The SBTB, a FIFO or circular buffer, allocates an entry when an instruction line containing a conditional branch is fetched and decoded, and deallocates it when the last branch in the line retires. The novel branch prediction is made based on the youngest (e.g., the most recently allocated or updated) of the entries in the ABTB or the SBTB. Branch allocation/deallocation is done at branch decode time on the SBTB, leaving the ABTB untouched. Speculative prediction is continuously made, assuming it is correct, for subsequent processing, until an actual entry is made in the architectural history. Further, any mispredicted entries are corrected at execution time on the SBTB, and branch update is done on the ABTB at retirement. The method is designed to reduce the cost of branch prediction and increase its performance. Hence, producing an efficient, yet affordable, microprocessor.

In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form.

Importantly, the method and apparatus of the present invention conceptually operate at a layer above branch prediction. Therefore, while embodiments of the present invention will be described with reference to branch prediction algorithms employing pattern tables, the method and apparatus described herein are equally applicable to other branch prediction techniques, such as the Yeh algorithm (See Tse Yu Yeh and Yale N. Patt, "Two-Level Adaptive Branch Prediction," The 24th ACM/IEEE International

Symposium and Workshop on Microarchitecture, November 1991, pp. 51-61), and other static and dynamic branch prediction mechanisms.

Computer System Overview

Figure 4A illustrates a computer system 400 representing an exemplary target system in which features of the present invention may be implemented. Computer system 400 comprises a bus or other communication means 401 for communicating information, and a processing means such as processor 402 coupled with bus 401 for processing information. The processor 402 comprises a novel branch prediction circuit 403 that will be described further below.

Computer system 400 further comprises a random access memory (RAM) or other dynamic storage device 404 (referred to as main memory), coupled to bus 401 for storing information and instructions to be executed by processor 402. Main memory 404 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 402. Computer system 400 also comprises a read only memory (ROM) and/or other static storage device 406 coupled to bus 401 for storing static information and instructions for processor 402.

A data storage device 407 such as a magnetic disk or optical disc and its corresponding drive may also be coupled to computer system 400 for storing information and instructions. Computer system 400 can also be coupled via bus 401 to a display device 421, such as a cathode ray tube (CRT) or Liquid Crystal Display (LCD), for displaying information to an end user. For example, graphical and/or textual indications of installation status, time remaining in the trial period, and other information may be presented to the prospective purchaser on the display device 421. Typically, an alphanumeric input device

422, including alphanumeric and other keys, may be coupled to bus 401 for communicating information and/or command selections to processor 402. Another type of user input device is cursor control 423, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 402 and for controlling cursor movement on display 421.

A communication device 425 is also coupled to bus 401. The communication device 425 may include a modem, a network interface card, or other well-known interface devices, such as those used for coupling to Ethernet, token ring, or other types of physical attachment for purposes of providing a communication link to support a local or wide area network, for example. In any event, in this manner, the computer system 400 may be coupled to a number of clients and/or servers via a conventional network infrastructure, such as a company's Intranet and/or the Internet, for example.

It is appreciated that a lesser or more equipped computer system than the example described above may be desirable for certain implementations. Therefore, the configuration of computer system 400 will vary from implementation to implementation depending upon numerous factors, such as price constraints, performance requirements, technological improvements, and/or other circumstances.

It should be noted that, while the steps described herein may be performed under the control of a programmed processor, such as processor 402, in alternative embodiments, the steps may be fully or partially implemented by any programmable or hardcoded logic, such as Field Programmable Gate Arrays (FPGAs), TTL logic, or Application Specific Integrated Circuits (ASICs), for example. Additionally, the method of the present invention may be performed by any combination of programmed general purpose computer components and/or custom hardware components. Therefore, nothing

disclosed herein should be construed as limiting the present invention to a particular embodiment wherein the recited steps are performed by a specific combination of hardware components.

Figure 4B is a simplified block diagram of processor 402 including various units that may interact with the branch prediction circuit of the present invention. In this example, the processor 402 includes a fetch unit 410, a branch prediction circuit 420, a decode unit 430, an execution unit 440, a retirement unit 450, and a cache 460. The fetch unit 410 retrieves instructions from cache and uses the instruction pointer (IP) to continuously fetch based on the signals received from the branch prediction circuit 420. In this example, the branch prediction circuit 420 includes a novel branch target buffer (BTB) 470 comprising a speculative branch target buffer (SBTB) 490 and an architectural branch target buffer 480. The branch prediction circuit 420 identifies branches and predicts whether or they will be taken based upon branch data contained in the SBTB 490 and the ABTB 480 as will be described further below. The SBTB 490 includes a plurality of branch entries (not shown) to maintain speculative branch data associated with in-flight branches thereby reducing the burden on the ABTB 480 and allowing the ABTB 480 to track only architectural branch data, such as the actual (fully-resolved) taken/not-taken history associated with retired conditional branches.

Returning to the fetch unit 410, the fetching process of the fetch unit 410 is interrupted if a branch is encountered, because the next instruction following the branch needs to be resolved before further instructions can be fetched. The branch prediction circuit 420 predicts the target address of the branch instruction based upon whether or not the branch is predicted as taken. The branch prediction circuit 420 provides this address to the fetch unit 410 to allow the fetch unit 410 to continue fetching instruction data.

The predicted target address is forwarded to the decode unit 430. The decode unit 430 verifies each branch prediction and decodes each branch instruction. While verifying the results of the branch prediction, the decode unit 503 may deallocate any bogus branches that it detects. A bogus branch is one predicted by the branch prediction circuit 420 at a location where no branch instructions exist.

The execution unit 440 then executes the branch instruction. The execution unit 440 compares the predicted branch target with the actual branch target, and hence may determine whether the branch was correctly predicted. The execution unit 440 may corrects any mispredicted branches or mispredicted targets by flushing the head of the pipeline and updating the corresponding branch entry in the SBTB 490.

Finally, the retirement unit 450 retires each branch instruction. According to one embodiment, branch data may be updated at this point by stalling the prediction pipeline and writing back a line to the ABTB 480 when the last branch in the line retires. By updating branch data only when the last of the branches in the line has retired, update traffic to the ABTB 480 is reduced thus making it possible to implement the ABTB 480 as a single-ported cache. Additionally, branch updating during retirement eliminates BTB corruption that may result from prior art update mechanisms that attempt to update the BTB at execution time. While such update mechanisms may improve prediction, corruption of the BTB may result since not all executed branches actually retire.

Branch Prediction Circuit

Figure 5 is a simplified block diagram of a branch prediction circuit 500 according to one embodiment of the present invention. In the embodiment depicted, the branch prediction circuit 500 includes an architectural branch target buffer (ABTB) 510, a

speculative branch target buffer (SBTB) 520, and selection logic 530. According to one embodiment, the SBTB 520 is a relatively small structure supporting the ABTB 510. The SBTB 520 is used to maintain the speculative branch data for in-flight branches, meaning fetched branches that are not yet retired. In the embodiment depicted, the SBTB includes an N stage FIFO, where N is the number of stages in the processor's instruction pipeline and a branch allocation register 523. Each stage of the FIFO includes per-line fields 521 and per-way fields 522.

Per-line fields 521 include a set field, a pattern table, least recently replaced (LRR) bits, a BAR index, and a sequential set indication. The set field identifies the set number. In this manner, all entries of the SBTB 520 may search in parallel to see whether the set matches the IP. The pattern table is typically updated at retirement. However, it may be updated at prediction if deemed worthwhile for prediction accuracy. The LRR bits point to the entry to be replaced if necessary. Preferably, entries outside the line, or outside the execution path are selected if possible. The BAR indication indicates the branch allocation register used for allocation or that there is no allocation. If there is an allocation, the LRR bits indicate the entry being replaced. This is used for any subsequent predictions. The sequential set indication indicates whether the next set is a sequential set. This is used to deallocate entries in the next set in the case of a bogus branch.

Per-way fields 522 include a valid indication, an order field, a speculative bit, history information, and a prediction field. The valid indication indicates whether or not the branch is valid. This bit is set on allocation and cleared on deallocation. The order field indicates the order of the branch offsets from lowest to highest. The speculative bit indicates that the branch was speculatively updated. This bit is cleared when updated at

retirement. It is also used to deallocate the line when the last branch is updated. History information contains the latest history copies from the ABTB or the SBTB. This allows the pattern table to be updated at retirement. Finally, the prediction bit represents the prediction. The prediction bit is concatenated with the last 3 history bits to form the history to be used for the next prediction.

Branch allocation registers each include an indication of the type of branch being allocated, the tag of the branch, the offset of the branch, and history to be initialized based upon the type.

Because the SBTB 520 is read/written during the decode stage for allocation of branch entries, during the execution stage for speculative update of branch entries, and during the retirement stage to correct branch entries, it is preferable to implement the SBTB 520 as a dual-ported memory.

The ABTB 510 need only be read during branch prediction and written when branches in the SBTB 520 have retired. Consequently, the ABTB 510 may be implemented with a single read port and a single write port. Alternatively, the ABTB 510 may be implemented as a single-ported memory in which reading and writing occur over the same shared port.

Selection logic 530 selects between the ABTB output and the SBTB output, depending upon which one of the two contains the youngest entry.

Branch Entry Processing

Figure 6is a flow diagram illustrating branch entry processing according to one embodiment of the present invention. When no entry is found in the ABTB 601, an entry is allocated in the SBTB at decode time 603. In case of a bogus branch 604, deallocation

is performed at decode time 605, else the branch is predicted speculatively 606. The speculative prediction continuous 606, assuming the prediction is correct, for the subsequent entries until an actual entry in found in the ABTB 607. Once there is an actual entry in the ABTB, any corresponding entry in the SBTB is decoded in order to avoid duplication 608. Any mispredicted branches and mispredicted targets are corrected at execute time, and entries are later executed 610. Finally, the executed branch instructions are retired 611. The branch history and PT are updated, but only branches that actually retire update the ABTB. Since not all executed branches retire, branch update at Retire time eliminates corruption.

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is

1. A method comprising maintaining speculative branch data for in-flight branches in a speculative branch target buffer (SBTB) cache by speculatively allocating a branch entry in a line of the SBTB after decoding an instruction containing a branch, speculatively updating branch data associated with the branch entry after branch prediction has been completed for the branch, and correcting the branch data after the branch has been executed.

2. The method of claim 1, wherein the branch data includes a speculative history field representing the speculative taken or not-taken history of the branch for a predetermined window of executions of the branch, and wherein said speculatively updating branch data comprises updating the speculative history field to reflect the taken or not-taken status of its most recent execution.

3. The method of claim 1, wherein the line has a corresponding a pattern table, and wherein said speculatively updating branch data comprises updating the pattern table.

4. The method of claim 1, wherein the branch comprises a conditional branch.

5. The method of claim 1, wherein the branch comprises a return from a subroutine.

6. The method of claim 1, wherein the branch comprises a call to a subroutine.

7. The method of claim 1, wherein the branch comprises an unconditional branch.

8. A method comprising:

speculatively allocating a first branch entry for a conditional branch in a speculative branch target buffer (SBTB) prior to execution of the conditional branch responsive to decoding the conditional branch and finding no branch entry in an architectural branch target buffer (ABTB) corresponding to the conditional branch;

speculatively allocating a second branch entry for the conditional branch in a the SBTB responsive to a subsequent failed attempt to locate a branch entry in the ABTB corresponding to the conditional branch;

allocating a third branch entry for the conditional branch in the ABTB after retirement of the conditional branch; and

subsequently performing branch prediction for the conditional branch by determining a predicted target address branch based upon branch data associated with the second branch entry.

9. The method of claim 8, further comprising speculatively updating branch data associated with the first branch entry after said performing branch prediction for the conditional branch.

10. A branch prediction circuit comprising:

speculative branch target buffer (SBTB) means for maintaining speculative branch data associated with in-flight branches;

architectural branch target buffer (ABTB) means, coupled to the SBTB means, for maintaining architectural branch data for branches corresponding to retired instructions; and

target address generation means coupled to both the SBTB means and the ABTB means for determining a predicted target address based upon the speculative branch data and the architectural branch data.

11. The branch prediction circuit of claim 10, wherein the SBTB means comprises a FIFO having entries corresponding to each of a plurality of pipeline stages of a microprocessor instruction pipeline.

12. The branch prediction circuit of claim 10, wherein the SBTB means includes a single read port and a single write port.

13. The branch prediction circuit of claim 10, wherein the SBTB means comprises a single-ported memory.

14. A machine-readable medium having stored thereon data representing sequences of instructions, the sequences of instructions which, when executed by a processor, cause the processor to:

speculatively allocate a first branch entry for a conditional branch in a speculative branch target buffer (SBTB) prior to execution of the conditional branch responsive to decoding the conditional branch and finding no branch entry in an architectural branch target buffer (ABTB) corresponding to the conditional branch;

speculatively allocate a second branch entry for the conditional branch in a the SBTB responsive to a subsequent failed attempt to locate a branch entry in the ABTB corresponding to the conditional branch;

allocate a third branch entry for the conditional branch in the ABTB after retirement of the conditional branch; and

subsequently perform branch prediction for the conditional branch by determining a predicted target address branch based upon branch data associated with the second branch entry.

15. The machine-readable medium of claim 14, wherein the sequences of instructions further cause the processor to speculatively update branch data associated with the first branch entry after said performing branch prediction for the conditional branch.

16. A branch prediction circuit comprising:

a speculative branch target buffer (SBTB) cache having a plurality of branch entries to maintain speculative branch data associated with in-flight branches, the speculative branch data including a speculative history of taken/not-taken outcomes associated with the in-flight branches; and

an architectural branch target buffer (ABTB) cache, coupled to the SBTB cache, the ABTB cache having a plurality of branch entries to maintain architectural branch data including the actual taken/not-taken outcomes associated with retired conditional branches.

17. The branch prediction circuit of claim 16, wherein the SBTB cache comprises a FIFO having entries corresponding to each of a plurality of pipeline stages of a microprocessor instruction pipeline.

18. The branch prediction circuit of claim 16, wherein the SBTB cache is dual-ported.

19. The branch prediction circuit of claim 16, wherein the SBTB cache is single-ported.

20. The branch prediction circuit of claim 16, wherein the ABTB cache is single-ported.

21. A processor comprising:

a fetch unit to speculatively retrieve instruction data for processing by an instruction pipeline; and

a branch prediction circuit, coupled to the fetch unit, to predict final target addresses for branch instructions contained within the instruction data, the branch prediction circuit including

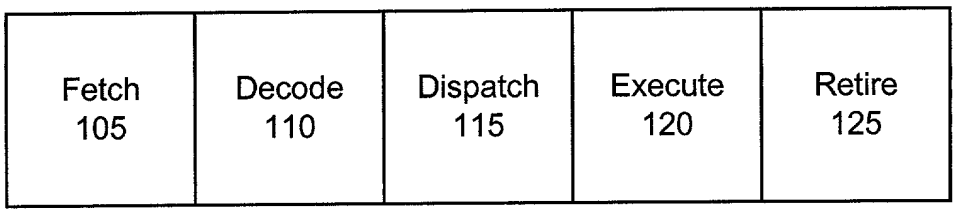
a speculative branch target buffer (SBTB) cache having a plurality of branch entries to maintain speculative branch data associated with in-flight branches, the speculative branch data including a speculative history of taken/not-taken outcomes associated with the in-flight branches, and

an architectural branch target buffer (ABTB) cache, coupled to the SBTB cache, the ABTB having a plurality of branch entries to maintain architectural branch data including the actual taken/not-taken outcomes associated with retired conditional branches.

22. The processor of claim 21, wherein the SBTB cache comprises a FIFO having entries corresponding to each of a plurality of pipeline stages of the instruction pipeline.

ABSTRACT OF THE DISCLOSURE

A method and apparatus are provided for improving the performance of branch prediction using a combination of a speculative branch target buffer (SBTB) and an architectural branch target buffer (ABTB). According to one embodiment, speculative branch data is maintained for in-flight branches (i.e., those that have been fetched but not yet retired). A branch entry is speculatively allocated in a line of the SBTB after decoding an instruction containing a branch, such as a conditional branch, a return from a subroutine, a call to a subroutine, or an unconditional branch. Subsequently, the branch data associated with the branch entry is speculatively updated after branch prediction has been completed for the branch. Finally, the branch data is corrected after the branch has been executed. According to another embodiment, a novel branch prediction circuit includes both a speculative branch target buffer (SBTB) cache and an architectural branch target buffer (ABTB) cache. The SBTB cache contains multiple branch entries to maintain speculative branch data associated with in-flight branches. The speculative branch data includes a speculative history of taken/not-taken outcomes associated with the in-flight branches. The ABTB cache is coupled to the SBTB cache. The ABTB cache also includes multiple branch entries, however, they are for maintaining architectural branch data including the actual taken/not-taken outcomes associated with retired conditional branches.



100

Figure 1

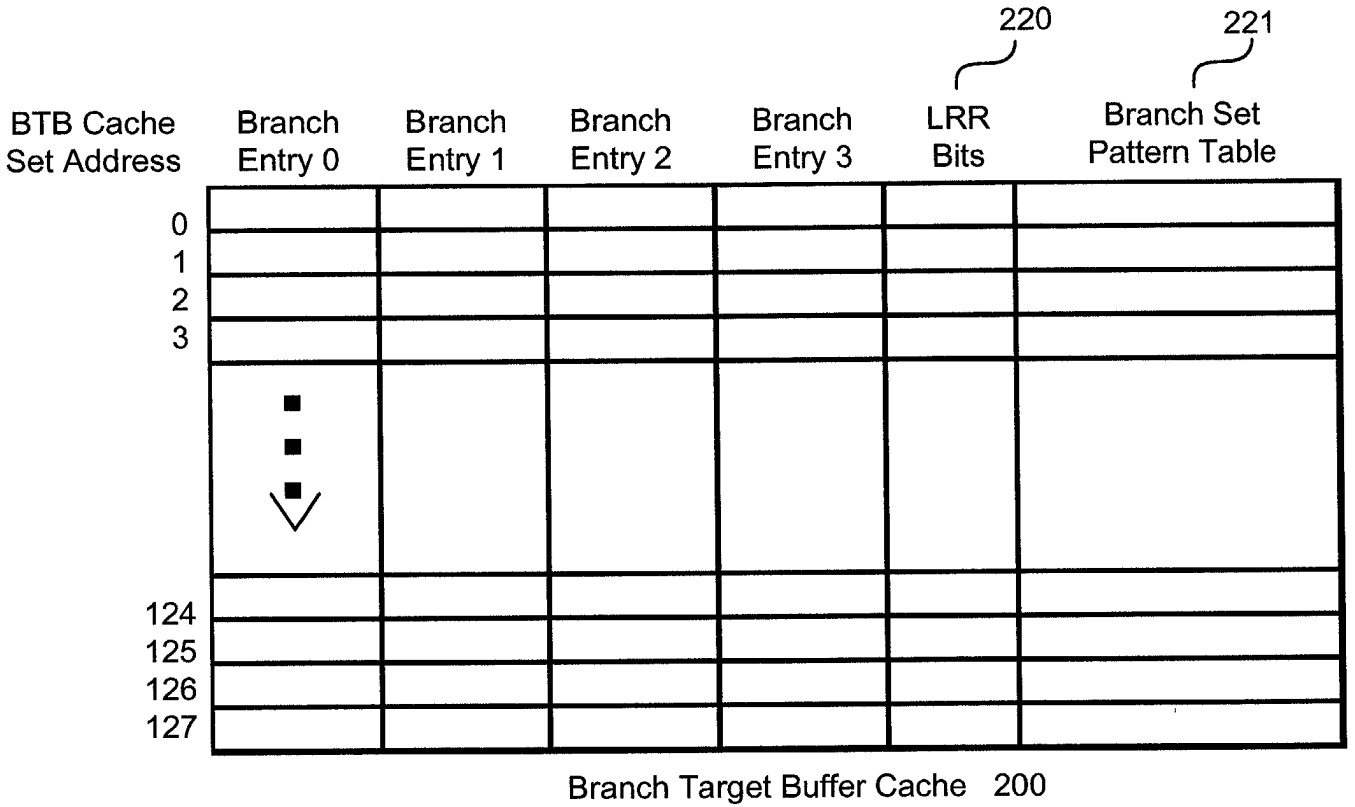


Figure 2 (Prior Art)

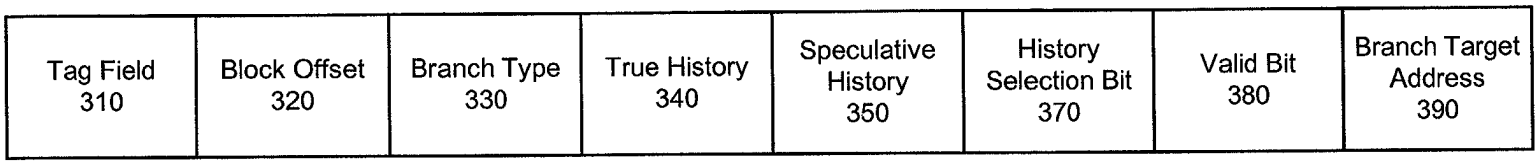


Figure 3

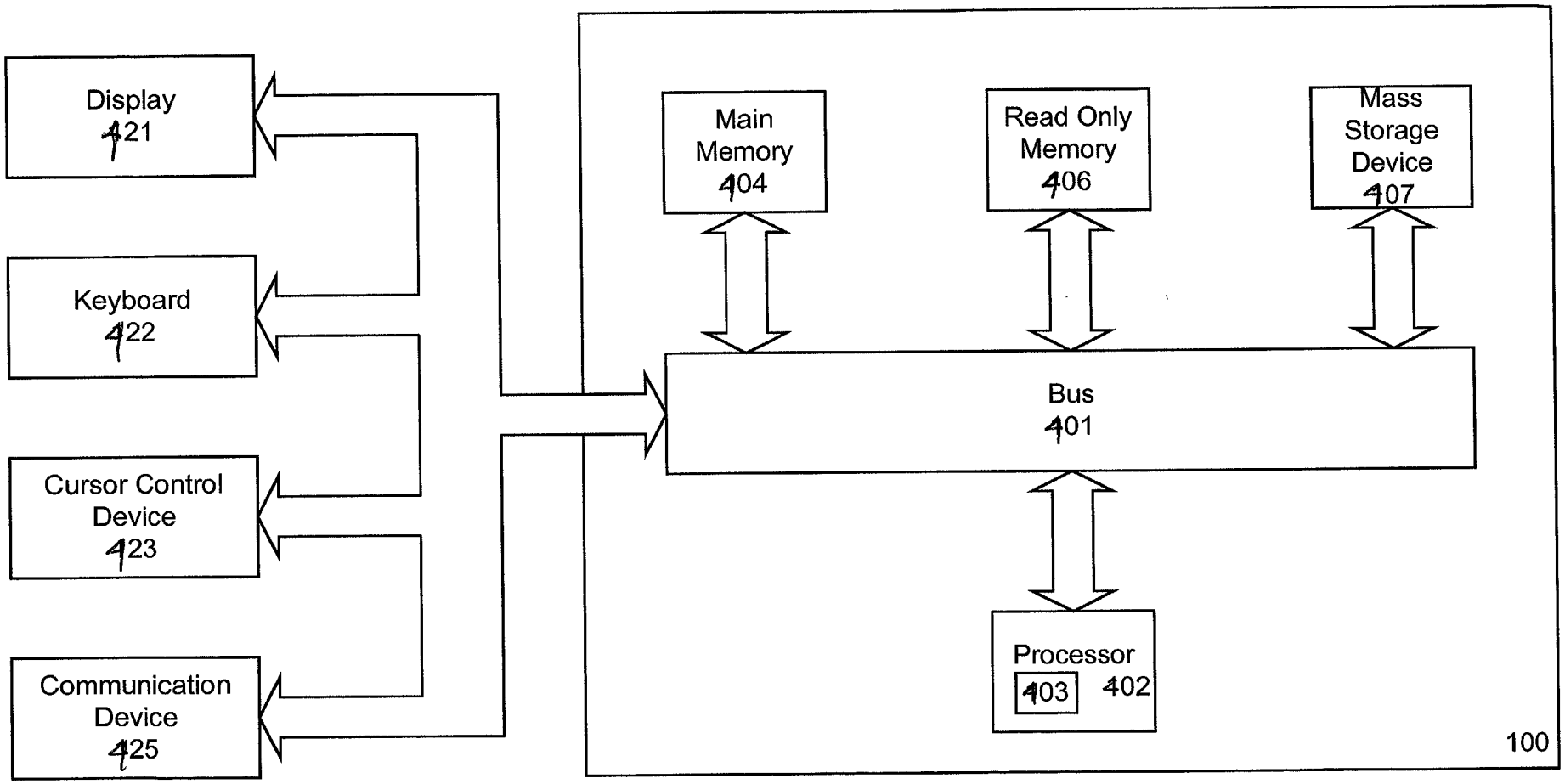


Figure 4A

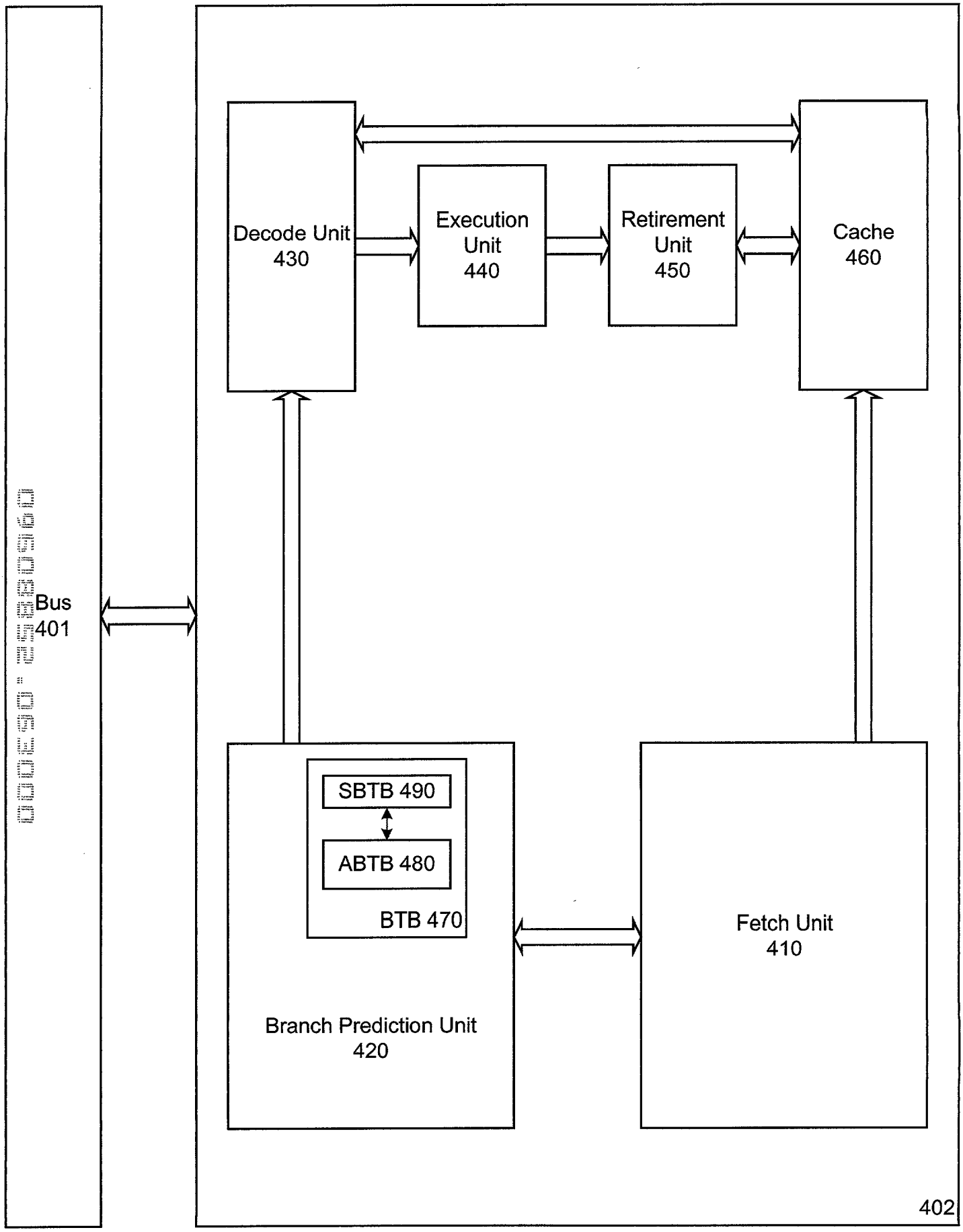

Bus
401
Decode Unit
430
Execution
Unit
440
Retirement
Unit
450
Cache
460
SBTB 490
ABTB 480
BTB 470
Branch Prediction Unit
420
Fetch Unit
410
402

Figure 4B

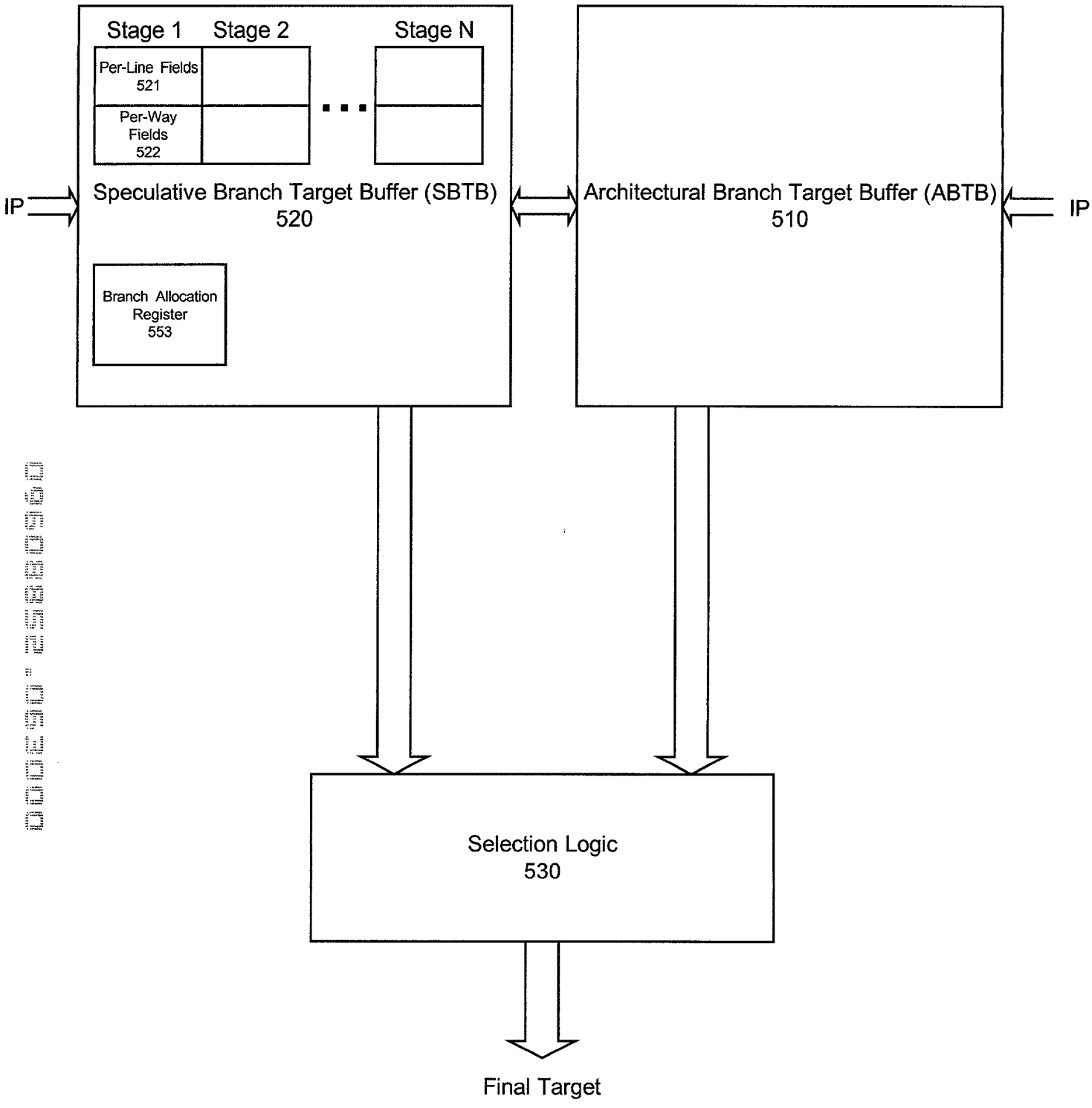

Stage 1
Stage 2
Stage N
Per-Line Fields
521
Per-Way
Fields
522
IP
Speculative Branch Target Buffer (SBTB)
520
Branch Allocation
Register
553
Architectural Branch Target Buffer (ABTB)
510
IP
Selection Logic
530
Final Target

Figure 5

500

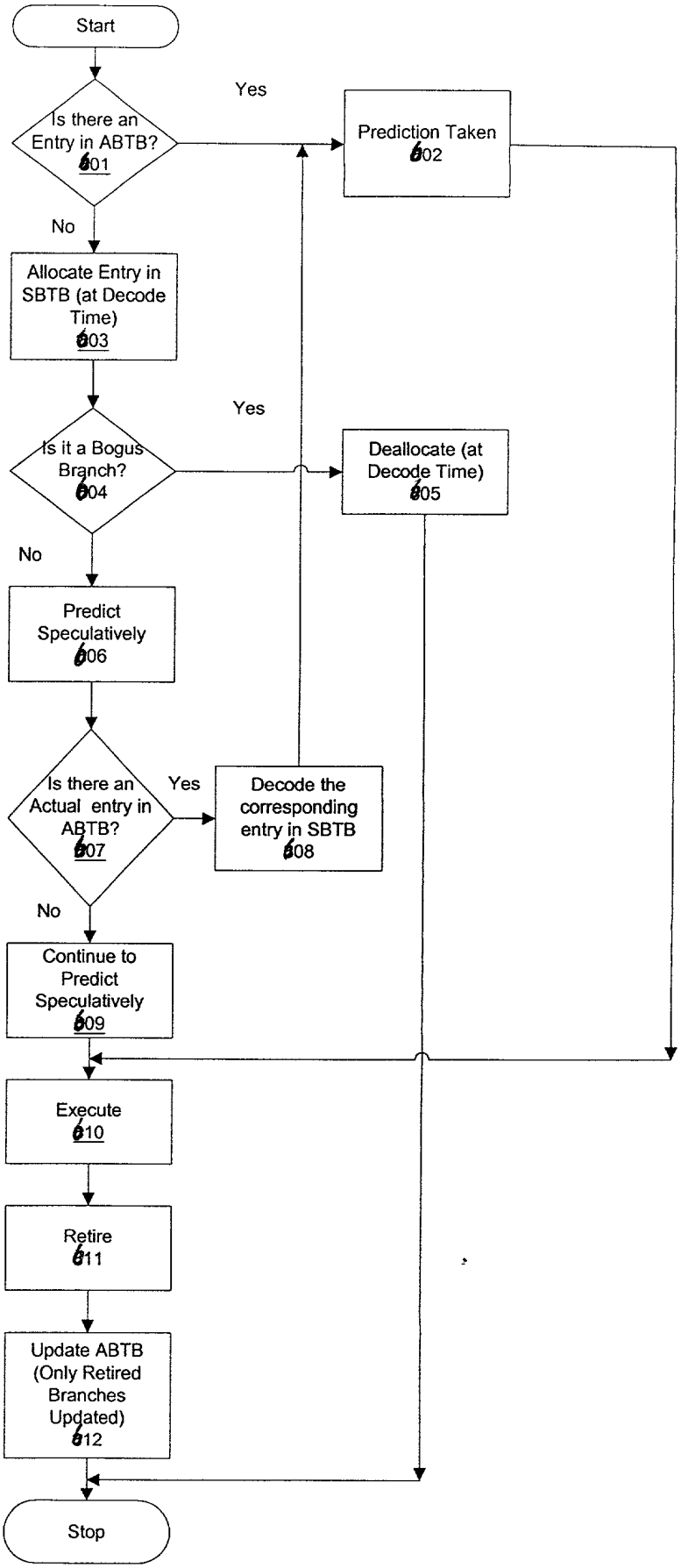

Start
Is there an Entry in ABTB? 601
Yes
Prediction Taken 602
No
Allocate Entry in SBTB (at Decode Time) 603
Is it a Bogus Branch? 604
Yes
Deallocate (at Decode Time) 605
No
Predict Speculatively 606
Is there an Actual entry in ABTB? 607
Yes
Decode the corresponding entry in SBTB 608
No
Continue to Predict Speculatively 609
Execute 610
Retire 611
Update ABTB (Only Retired Branches Updated) 612
Stop

Fig 6.

Attorney's Docket No.: 042390.P5563 PATENT

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION
(FOR **INTEL CORPORATION** PATENT APPLICATIONS)

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

A Branch Target Buffer (BTB) including a Speculative BTB (SBTB) and an Architectural BTB (ABTB)

the specification of which

X is attached hereto.
___ was filed on _____ as
United States Application Number _____
or PCT International Application Number _____
and was amended on _____.
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

Priority Claimed

(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below:

Application Number	Filing Date
Application Number	Filing Date

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

Application Number	Filing Date	Status -- patented, pending, abandoned
Application Number	Filing Date	Status -- patented, pending, abandoned

I hereby appoint the persons listed on Appendix A hereto (which is incorporated by reference and a part of this document) as my respective patent attorneys and patent agents, with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

Send correspondence to Michael A. DeSanctis, BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, 12400 Wilshire Boulevard 7th Floor, Los Angeles, California 90025 and direct telephone calls to (303) 740-1980.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor Kiran A. Padwekar

Inventor's Signature _____ Date _____

Residence Santa Clara, CA (City, State) Citizenship United States of American (Country)

Post Office Address 1520 Vista Club Circle, #203
Santa Clara, CA 95054

Full Name of Second/Joint Inventor _____

Inventor's Signature _____ Date _____

Residence _____ (City, State) Citizenship _____ (Country)

Post Office Address _____

Full Name of Third/Joint Inventor _____

Inventor's Signature _____ Date _____

Residence _____ (City, State) Citizenship _____ (Country)

Post Office Address _____

APPENDIX A

William E. Alford, Reg. No. 37,764; Farzad E. Amini, Reg. No. P42,261; Aloysius T. C. AuYeung, Reg. No. 35,432; William Thomas Babbitt, Reg. No. 39,591; Carol F. Barry, Reg. No. 41,600; Jordan Michael Becker, Reg. No. 39,602; Lisa N. Benado, Reg. No. 39,995; Bradley J. Bereznak, Reg. No. 33,474; Michael A. Bernadicou, Reg. No. 35,934; Roger W. Blakely, Jr., Reg. No. 25,831; R. Alan Burnett, Reg. No. 46,149; Gregory D. Caldwell, Reg. No. 39,926; Andrew C. Chen, Reg. No. 43,544; Thomas M. Coester, Reg. No. 39,637; Donna Jo Coningsby, Reg. No. 41,684; Florin Corie, Reg. No. 46,244; Dennis M. deGuzman, Reg. No. 41,702; Stephen M. De Klerk, Reg. No. P46,503; Michael Anthony DeSanctis, Reg. No. 39,957; Daniel M. De Vos, Reg. No. 37,813; Robert Andrew Diehl, Reg. No. 40,992; Sanjeet Dutta, Reg. No. P46,145; Matthew C. Fagan, Reg. No. 37,542; Tarek N. Fahmi, Reg. No. 41,402; George Fountain, Reg. No. 37,374; Paramita Ghosh, Reg. No. 42,806; James Y. Go, Reg. No. 40,621; Libby N. Ho, Reg. No. P46,774; James A. Henry, Reg. No. 41,064; Willmore F. Holbrow III, Reg. No. P41,845; Sheryl Sue Holloway, Reg. No. 37,850; George W Hoover II, Reg. No. 32,992; Eric S. Hyman, Reg. No. 30,139; William W. Kidd, Reg. No. 31,772; Sang Hui Kim, Reg. No. 40,450; Walter T. Kim, Reg. No. 42,731; Eric T. King, Reg. No. 44,188; Erica W. Kuo, Reg. No. 42,775; George Brian Leavell, Reg. No. 45,436; Kurt P. Leyendecker, Reg. No. 42,799; Gordon R. Lindeen III, Reg. No. 33,192; Jan Carol Little, Reg. No. 41,181; Joseph Lutz, Reg. No. 43,765; Michael J. Mallie, Reg. No. 36,591; Andre L. Marais, under 37 C.F.R. § 10.9(b); Paul A. Mendonsa, Reg. No. 42,879; Clive D. Menezes, Reg. No. 45,493; Chun M. Ng, Reg. No. 36,878; Thien T. Nguyen, Reg. No. 43,835; Thinh V. Nguyen, Reg. No. 42,034; Dennis A. Nicholls, Reg. No. 42,036; Daniel E. Ovanezian, Reg. No. 41,236; Kenneth B. Paley, Reg. No. 38,989; Marina Portnova, Reg. No. P45,750; William F. Ryann, Reg. 44,313; James H. Salter, Reg. No. 35,668; William W. Schaal, Reg. No. 39,018; James C. Scheller, Reg. No. 31,195; Jeffrey Sam Smith, Reg. No. 39,377; Maria McCormack Sobrino, Reg. No. 31,639; Stanley W. Sokoloff, Reg. No. 25,128; Judith A. Szepesi, Reg. No. 39,393; Vincent P. Tassinari, Reg. No. 42,179; Edwin H. Taylor, Reg. No. 25,129; John F. Travis, Reg. No. 43,203; Joseph A. Twarowski, Reg. No. 42,191; Tom Van Zandt, Reg. No. 43,219; Lester J. Vincent, Reg. No. 31,460; Glenn E. Von Tersch, Reg. No. 41,364; John Patrick Ward, Reg. No. 40,216; Mark L. Watson, Reg. No. P46,322; Thomas C. Webster, Reg. No. P46,154; Steven D. Yates, Reg. No. 42,242; and Norman Zafman, Reg. No. 26,250; my patent attorneys, and Firasat Ali, Reg. No. 45,715; and Justin M. Dillon, Reg. No. 42,486; my patent agents, of BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, with offices located at 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025, telephone (310) 207-3800, and Alan K. Aldous, Reg. No. 31,905; Edward R. Brake, Reg. No. 37,784; Ben Burge, Reg. No. 42,372; Jeffrey S. Draeger, Reg. No. 41,000; Cynthia Thomas Faatz, Reg No. 39,973; John N. Greaves, Reg. No. 40,362; Seth Z. Kalson, Reg. No. 40,670; David J. Kaplan, Reg. No. 41,105; Peter Lam, Reg. No. 44,855; Charles A. Mirho, Reg. No. 41,199; Leo V. Novakoski, Reg. No. 37,198; Thomas C. Reynolds, Reg. No. 32,488; Kenneth M. Seddon, Reg. No. 43,105; Mark Seeley, Reg. No. 32,299; Steven P. Skabrat, Reg. No. 36,279; Howard A. Skaist, Reg. No. 36,008; Gene I. Su, Reg. No. 45,140; Calvin E. Wells, Reg. No. P43,256, Raymond J. Werner, Reg. No. 34,752; Robert G. Winkle, Reg. No. 37,474; and Charles K. Young, Reg. No. 39,435; my patent attorneys, of INTEL CORPORATION; and James R. Thein, Reg. No. 31,710, my patent attorney with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

APPENDIX B

Title 37, Code of Federal Regulations, Section 1.56
Duty to Disclose Information Material to Patentability

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclosure information exists with respect to each pending claim until the claim is cancelled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is cancelled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclosure all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) Prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) The closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made or record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application;

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.